

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Cancelled)
2. (Previously Presented) A method according to claim 11, wherein the step of synchronizing includes the step of:  
blocking a commit of the transaction until the number of transactions in the buffers is in a predetermined numerical relationship with the predetermined number of transactions.
3. (Previously Presented) A method according to claim 2, wherein:  
said blocking the commit of the transaction until the number of transactions in the buffers is in the predetermined numerical relationship with the predetermined number of transactions including blocking the commit of the transaction until the number of transactions in the buffers is less than the predetermined number of transactions.
4. (Cancelled)
5. (Previously Presented) A method according to claim 11, wherein:  
the log writer process performs the step of synchronizing.
6. (Previously Presented) A method according to claim 11, wherein:  
a database application process performs the step of synchronizing before submitting the transaction to the log writer process.
7. (Cancelled)

8. (Previously Presented) A method according to claim 11, further comprising the steps of:
- receiving input from an operator indicating a transaction loss bound; and
  - setting the predetermined number of transactions based on the transaction loss bound.
9. (Cancelled)
10. (Currently Amended) A computer-readable medium bearing instructions for causing one or more processors to ~~performs~~ perform the steps of the method according to claim 11.
11. (Currently Amended) A method for replicating data of a primary database system, comprising the steps of:
- maintaining a queue of information associated with transactions to be sent to a standby database system;
  - storing a counter indicating a number of the transactions having information in the queue;
  - storing a predetermined bound of transactions having information in the queue;
  - executing a log writer process to:
    - record ~~[[the]]~~ a transaction in a redo log,
    - compare the counter and the predetermined bound,
    - if the counter is not less than the predetermined bound, then block a commit of the ~~transactions~~ transaction until the counter is less than the predetermined bound, and
    - if the counter is less than the predetermined bound, then increment the counter and acknowledge the commit of the transaction; and
  - executing a net server process to:
    - transmit the information associated with a transaction over a network connection to the standby database system,

receive an acknowledgement that a redo record based on the information associated with [[for]] the transaction has been written to a standby log at the standby database system, and

in response to the acknowledgement, remove the information associated with the transaction from the queue and decrement the counter.

12-15. (Cancelled)

16. (Previously Presented) A method for replicating data in a primary database system having multiple database servers operating in parallel and accessing a common database on a shared disk, said method comprising the steps of:

setting a bound for each of the multiple database servers;

for each of the multiple database servers, performing the steps of:

maintaining a buffer of transactions to be sent to a standby database system; and

synchronizing a transaction performed on the primary database system based on a number of transactions in the buffer and the corresponding bound;

storing a counter indicating a number of the transactions in the queue;

storing a predetermined bound of transactions;

executing a log writer process to:

record the transaction in a redo log,

compare the counter and the predetermined bound,

if the counter is not less than the predetermined bound, then block a commit of the transaction until the counter is less than the predetermined bound, and

if the counter is less than the predetermined bound, then increment the counter and acknowledge the commit of the transaction; and

executing a net server process to:

transmit the transaction over a network connection to the standby database system, receive an acknowledgement that a redo record for the transaction has been written to a standby log at the standby database system, and

in response to the acknowledgement, remove the transaction from the queue and decrement the counter.

17. (Previously Presented) The method according to claim 16, wherein the step of synchronizing includes the step of blocking a commit of the transaction until the number of transactions in the buffers is in a predetermined numerical relationship with the predetermined number of transactions.

18. (Previously Presented) The method according to claim 16, wherein said blocking the commit the transaction until the number of transactions in the buffer is in the predetermined numerical relationship with the predetermined number of transaction including blocking the commit of the transaction until the number of transactions in the buffer is less than the predetermined number of transactions.

19. (Previously Presented) The method according to claim 16, wherein the log writer process performs the step of synchronizing.

20. (Previously Presented) The method according to claim 16, wherein a database application process performs the step of synchronizing before submitting the transaction to the log writer process.

21. (Previously Presented) The method according to claim 16, further comprising the steps of:

receiving input from an operator indicating a transaction loss bound; and  
setting the predetermined number of transactions based on the transaction loss bound.

22. (Previously Presented) A computer-readable medium bearing instructions for causing one or more processors to perform the steps of the method according to claim 16.

23. (new) A method for replicating data of a first database system, comprising the steps of:

allowing transactions on the first database system to commit while a number of committed transactions on the first database system for which an acknowledgement that information relating to each transaction of the committed transactions has been stored at a second database system has not been received is less than a predetermined bound; and

blocking transactions on the first database system from committing while the number of committed transactions on the first database system for which an acknowledgement that information relating to each transaction of the committed transactions has been stored at the second database system has not been received is not less than the predetermined bound.

24. (new) The method of claim 23, wherein the information relating to each transaction includes redo information relating to the transaction.

25. (new) The method of claim 24, wherein the information relating to each transaction is stored at the second database system by writing the redo information to a redo log.

26. (new) The method of claim 23, further comprising:

sending information relating to each transaction of the committed transactions to second database system to be stored; and

receiving an acknowledgement that information relating to at least some of the committed transactions has been stored at the second database system.